

1 **CLAIMS**

2
3 1. A software architecture for a distributed computing system
4 comprising:

5 an application configured to handle requests submitted by remote devices
6 over a network; and

7 an application program interface to present functions used by the
8 application to access network and computing resources of the distributed
9 computing system, wherein the application program interface comprises a set of
10 classes that make available standards-based support for processing XML
11 documents.

12
13 2. A software architecture as recited in claim 1, wherein the set of
14 classes comprises:

15 an XmlReader class that enables non-cached forward only access to XML
16 data;

17 an XPathNavigator class that enables read-only random access to a data
18 store;

19 an XslTransform class that enables transforming of XML data using an
20 XSLT stylesheet;

21 a plurality of Xml Schema classes that enable constructing and editing of
22 schemas;

23 an XmlResolver class that enables resolving of external XML resources
24 named by a Uniform Resource Identifier (URI);
25

1 an XmlDocument class that enables structured data to be stored,
2 retrieved, and manipulated through a relational dataset; and

3 an XmlWriter class that enables a non-cached forward only way of
4 generating streams and files containing XML data.

5
6 3. A software architecture as recited in claim 2, wherein the set of
7 classes further comprises:

8 an XmlValidatingReader class that enables DTD, XDR and XSD schema
9 validation.

10
11 4. An XmlReader class of an application program interface, embodied
12 on one or more computer readable media that enables non-cached forward only
13 access to XML data, the XmlReader class comprising:

14 an XmlReader constructor that enables initialization of a new instance of
15 the XmlReader class; and

16 a Read method that enables reading of nodes of the XML data via the
17 XmlReader class instance.

18
19 5. An XmlReader class of an application program interface as recited in
20 claim 4, wherein the XmlReader class further comprises:

21 a BaseURI property that identifies a base URI of a current node of the
22 XML data; and

23 a NodeType property that identifies the type of the current node.
24
25

1 6. An XPathNavigator class of an application program interface,
2 embodied on one or more computer readable media, that enables read-only
3 random access to a data store, the XPathNavigator class comprising:

4 an XPathNavigator constructor that enables initialization of a new instance
5 of the XPathNavigator class;

6 a MoveToFirst method that enables moving to a first sibling of a current
7 node of XML data;

8 a MoveToNext method that enables moving to a next sibling of the current
9 node;

10 a MoveToPrevious method that enables moving to a previous sibling of the
11 current node;

12 a MoveToFirstChild method that enables moving to a first child of the
13 current node;

14 a MoveToParent method that enables moving to a parent of the current
15 node; and

16 a NodeType property that enables obtaining the type of the node that is
17 moved to.

18
19 7. An XslTransform class of an application program interface,
20 embodied on one or more computer readable media, that enables transforming of
21 XML data using an XSLT stylesheet, the XslTransform class comprising:

22 an XslTransform constructor that enables initialization of a new instance of
23 the XslTransform class;

24 a Load method that enables loading of the XSLT stylesheet; and
25

1 a Transform method that enables transforming of the specified XML data
2 using the loaded XSLT stylesheet and outputs the results.

3
4 **8.** A set of XmlSchema classes of an application program interface,
5 embodied on one or more computer readable media, that enable constructing and
6 editing of schemas, the set of XmlSchema classes comprising:

7 a Schema class that contains a definition of a schema;
8 a SchemaObject class that enables creating of an empty schema; and
9 a SchemaCollection class that contains a cache of defined XML Schema
10 Definition language (XSD) and XML-Data Reduced Language (XDR) schemas.

11
12 **9.** An XmlResolver class of an application program interface, embodied
13 on one or more computer readable media, that enables resolving of external XML
14 resources named by a Uniform Resource Identifier (URI), the XmlResolver class
15 comprising:

16 a ResolveURI method that enables resolving the absolute URI from a base
17 URI and a relative URI; and

18 a GetEntity method that enables mapping of the resolved URI to an object
19 containing identified resource.

10. An XmlDataDocument class of an application program interface, embodied on one or more computer readable media, that enables structured data to be stored, retrieved, and manipulated through a relational dataset, the XmlDataDocument class comprising:

a DataSet property that enables obtaining of a dataset that provides a relational representation of the data in a document;

a Load method that enables loading of the document using a specified data source and synchronizing the dataset with the loaded data.

11. An XmlWriter class of an application program interface, embodied on one or more computer readable media, that enables a non-cached forward only way of generating streams and files containing XML data, the XmlWriter class comprising:

an XmlWriter constructor that enables initialization of a new instance of the XmlWriter class; and

an WriteState property that enables obtaining of the state of an instance of the XmlWriter class; and

a plurality of Write methods that enable writing XML data via the instance of the XmlWriter class.

12. An XmlValidatingReader class of an application program interface, embodied on one or more computer readable media, that enables DTD, XDR and XSD schema validation, the XmlValidatingReader class comprising:

a ValidationType property that enables obtaining an indication of what type of validation to perform on a document;

1 a Read method that enables reading of nodes of the document so that
2 validation of the document can be performed.

3
4 **13.** An XmlValidatingReader class of an application program interface
5 as recited in claim 12, where the type of validation to perform is validation
6 according to DTD.

7
8 **14.** An XmlValidatingReader class of an application program interface
9 as recited in claim 12, where the type of validation to perform is validation
10 according to XSD schemas.

11
12 **15.** An XmlValidatingReader class of an application program interface
13 as recited in claim 12, where the type of validation to perform is validation
14 according to XDR schemas.

15
16 **16.** A distributed computer software architecture, comprising:
17 one or more applications configured to be executed on one or more
18 computing devices, the applications handling requests submitted from remote
19 computing devices;

20 a networking platform to support the one or more applications; and

21 an application programming interface to interface the one or more
22 applications with the networking platform, wherein the application program
23 interface comprises a set of classes that make available standards-based support
24 for processing documents written in a markup language.

1 17. A software architecture as recited in claim 16, wherein the set of
2 classes comprises:

3 an XmlReader class that enables non-cached forward only access to XML
4 data;

5 an XPathNavigator class that enables read-only random access to a data
6 store;

7 an XslTransform class that enables transforming of XML data using an
8 XSLT stylesheet;

9 a plurality of Xml Schema classes that enable constructing and editing of
10 schemas;

11 an XmlResolver class that enables resolving of external XML resources
12 named by a Uniform Resource Identifier (URI);

13 an XmlDataDocument class that enables structured data to be stored,
14 retrieved, and manipulated through a relational dataset; and

15 an XmlWriter class that enables a non-cached forward only way of
16 generating streams and files containing XML data.

17
18 18. A software architecture as recited in claim 17, wherein the set of
19 classes further comprises:

20 an XmlValidatingReader class that enables DTD, XDR and XSD schema
21 validation.

1 **19.** A computer system including one or more microprocessors and one
2 or more software programs, the one or more software programs utilizing an
3 application program interface to request services from an operating system, the
4 application program interface including separate commands to request services
5 that make available support for processing XML documents.

6
7 **20.** A software architecture as recited in claim 19, wherein the
8 commands include:

9 non-cached forward only access to XML data;

10 read-only random access to a data store;

11 transforming of XML data using an XSLT stylesheet;

12 constructing and editing of schemas;

13 resolving of external XML resources named by a Uniform Resource
14 Identifier (URI);

15 storage, retrieval, and manipulation of structured data through a relational
16 dataset; and

17 non-cached forward only generation of streams and files containing XML
18 data.

19
20 **21.** A software architecture as recited in claim 20, wherein the
21 commands further include:

22 validating DTD, XDR and XSD schemas.

1 **22.** A method comprising:

2 receiving one or more calls from one or more remote devices over a
3 network, wherein the one or more calls are to one or more functions that make
4 available support for processing XML documents; and
5 performing the requested XML document processing.

6
7 **23.** A software architecture as recited in claim 22, wherein the one or
8 more functions comprises:

9 an XmlReader function that enables non-cached forward only access to
10 XML data;

11 an XPathNavigator function that enables read-only random access to a data
12 store;

13 an XslTransform function that enables transforming of XML data using an
14 XSLT stylesheet;

15 a plurality of Xml Schema functions that enable constructing and editing of
16 schemas;

17 an XmlResolver function that enables resolving of external XML resources
18 named by a Uniform Resource Identifier (URI);

19 an XmlDataDocument function that enables structured data to be stored,
20 retrieved, and manipulated through a relational dataset; and

21 an XmlWriter function that enables a non-cached forward only way of
22 generating streams and files containing XML data.

1 **24.** A software architecture as recited in claim 23, wherein the one or
2 more functions further comprises:

3 an XmlValidatingReader function that enables DTD, XDR and XSD
4 schema validation.

5
6 **25.** A method comprising:
7 calling, to one or more remote devices over a network, one or more
8 functions that make available support for processing XML documents;
9 receiving, from the one or more remote devices, a response to the calling.

10
11 **26.** A software architecture as recited in claim 25, wherein the one or
12 more functions comprises:

13 an XmlReader function that enables non-cached forward only access to
14 XML data;

15 an XPathNavigator function that enables read-only random access to a data
16 store;

17 an XslTransform function that enables transforming of XML data using an
18 XSLT stylesheet;

19 a plurality of Xml Schema functions that enable constructing and editing of
20 schemas;

21 an XmlResolver function that enables resolving of external XML resources
22 named by a Uniform Resource Identifier (URI);

23 an XmlDataDocument function that enables structured data to be stored,
24 retrieved, and manipulated through a relational dataset; and
25

an XmlWriter function that enables a non-cached forward only way of generating streams and files containing XML data.

27. A software architecture as recited in claim 26, wherein the one or more functions further comprises:

an `XmlValidatingReader` function that enables DTD, XDR and XSD schema validation.